

# Examen

## 105000119 - Programación para Sistemas 10MI-Grado en Matemáticas e Informática

Lenguajes y Sistemas Informáticos e Ingeniería de Software

ETSI Informáticos

Universidad Politécnica de Madrid

Curso 2016/2017 - Julio 2017

### Normas

- El examen puntúa sobre **12 puntos**.
- La duración total del mismo es de **45 minutos**.
- Se deberá tener el DNI o el carnet de la UPM en lugar visible.
- No olvidar rellenar **apellidos, nombre y número de matrícula** en cada hoja.
- La solución al examen se proporcionará antes de la revisión.

### Cuestionario

- (1 punto) 1. Indique cómo mostrar en pantalla el código o 'status' devuelto por un comando o script en Unix tras su ejecución.

**Solución:**

**echo** \$?

- (1 punto) 2. Indique dos comandos básicos para mostrar manuales y documentación de ayuda de comandos Unix y Bash.

**Solución:** man help

- (1 punto) 3. Sea el siguiente script 'miscript.bash' en Bash:

```
#!/bin/bash
echo $#
echo $2
```

Escriba qué aparecerá en pantalla al lanzar dicho script de la siguiente forma:

`./miscript.bash hola y adios`

Nota: suponga que el script se ejecuta normalmente y que no hay problemas de permisos, etc.

**Solución:**

3

y

(1 punto) 4. Sea el siguiente extracto de comandos Bash en Unix:

```
if [ -d /tmp ]; then  
    cd /tmp || echo A  
else  
    echo B  
fi  
echo C
```

Suponiendo que el directorio `/tmp` existe y que se dispone de los permisos adecuados para poder ejecutar con éxito el comando `cd /tmp`, escriba la salida que mostrará la ejecución de dicho extracto de código.

Nota: el comando `'[-d /tmp ]'` se puede escribir de forma equivalente como: `'test -d /tmp'`.

**Solución:** C

- (1 punto) 5. Dado la siguiente parte de código de un programa en C

```
struct fecha
{
    int dia;
    int mes;
    int anyo;
};
struct fecha hoy;
struct fecha *puntero;
puntero = &hoy;
```

Escribe la sentencia para almacenar en el campo 'mes' el valor 7 utilizando 'puntero'.

**Solución:** Cualquiera de las dos siguientes:

```
puntero->mes=7
(*puntero).mes=7
```

- (1 punto) 6. Escribe el código necesario para inicializar todo los elementos de 'matriz' a cero

```
int matriz[300][300];
```

**Solución:**

```
for( a = 0; a < 300; a=a+1 )
    for( b = 0; b < 300; b=b+1 )
        matriz[a][b] = 0;
```

- (1 punto) 7. ¿ Qué fichero cabecera hay que incluir si en nuestro programa hay que usar las funciones malloc() y calloc() ?

**Solución:**

```
#include <stdlib.h>
```

- (1 punto) 8. Dado el siguiente fragmento de código, ¿ Cuántas veces se imprime la palabra 'Hola' ?

```
#include<stdio.h>
int main()
{
    int x;
    for(x=-1; x<=10; x++)
    {
        if(x < 5) {
```

```
        continue;

    } else {
        break;
    }
    printf("Hola\n");
}
return 0;
}
```

- A. Infinitas
- B. 11 veces
- C. Ninguna
- D. 5 veces

**Solución:** C. Ninguna

- (2 puntos) 9. Indicar dos diferencias de comportamiento entre las funciones malloc() y realloc() Pueden ser sobre su comportamiento, sobre cómo se hacen las llamadas a esas funciones, etc.

**Solución:** malloc() debe ser llamada con parámetro (tamaño) y realloc() con dos (puntero y tamaño)

realloc() permite modificar el tamaño de memoria reservada anteriormente por una llamada a malloc() o calloc() y malloc() no lo puede hacer.

- (2 puntos) 10. Escribir todo el código fuente de un programa en C que imprima por pantalla los argumentos de la línea de comandos con los que se ha llamado.

**Solución:**

```
#include <stdio.h>
int main(int argc, char *argv[])
int i;
for (i=1; i<argc, i++)    printf("%s\n", argv[i]);
return 0;
```